Improving regularized singular value decomposition for collaborative filtering

Arkadiusz Paterek <paterek@mimuw.edu.pl>
Institute of Informatics, Warsaw University

August 12, 2007

7.04%

improvement over Netflix's recommendation system

The goal

- to predict users' preferences for movies, as accurately as possible

Outline of the approach

- take many well predicting methods
- combine their results with linear regression on the *test* set

The most important methods in the ensemble

- regularized SVD - by Simon Funk
- *New:* regularized SVD with biases
- clustering users using K-means
- postprocessing SVD results with item-item K-nearest neighbors
- *New:* postprocessing SVD with kernel ridge regression
- *New:* weighted linear model for each item
- *New:* methods similar to SVD, but with fewer parameters

Together they give 6.34% improvement.
The 7.04% solution is a result of linear regression of 56 predictors and 63 two-way interactions between them, plus partial cross-validation.

Regularized SVD

Predictions for user i and movie j:

$$\hat{y}_{ij} = u_i^{\mathsf{T}} v_j$$

where u_i and v_j are K-dimensional vectors of parameters.

Parameters are estimated by minimizing the sum of squared residuals, one feature at a time, using gradient descent with regularization and early stopping:

$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

$$u_{ik} \mathrel{+}= lrate * (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} \mathrel{+}= lrate * (r_{ij} u_{ik} - \lambda v_{jk})$$

Regularized SVD with biases

We add biases to the regularized SVD model, one parameter c_i for each user and one d_j for each movie:

$$\hat{y}_{ij} = c_i + d_j + u_i^{\mathsf{T}} v_j$$

Weights c_i, d_j are trained simultaneously with u_{ik} and v_{jk}:

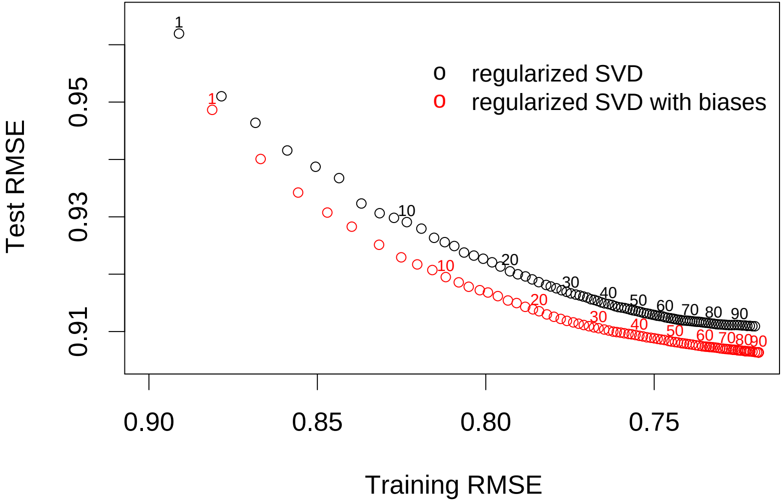
$$r_{ij} = y_{ij} - \hat{y}_{ij}$$

$$u_{ik} \mathrel{+}= lrate * (r_{ij} v_{jk} - \lambda u_{ik})$$

$$v_{jk} \mathrel{+}= lrate * (r_{ij} u_{ik} - \lambda v_{jk})$$

$$c_i \mathrel{+}= lrate * (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$

$$d_j \mathrel{+}= lrate * (r_{ij} - \lambda_2 (c_i + d_j - \mu))$$



o regularized SVD
o regularized SVD with biases
Test RMSE
Training RMSE
0.91
0.93
0.95
0.90
0.85
0.80
0.75

Postprocessing SVD with kernel ridge regression

- idea: discard all weights u_{ik} after training and, for each user i, predict y_{ij} using $x_{j_2} = \frac{v_j}{||v_j||}$ as predictors
- using ridge regression gives similar results to the reg. SVD
- better prediction can be obtained by using kernel ridge regression with Gaussian kernel
- predictions in kernel ridge regression:

$$\hat{y}_j = K(x_j^{\mathsf{T}}, X)(K(X, X) + \lambda I)^{-1} y$$

- in place of the scalar product we use a Gaussian kernel:

$$K(x_j^{\mathsf{T}}, x_k^{\mathsf{T}}) = exp(2(x_j^{\mathsf{T}} x_k - 1))$$

Weighted linear model for each item

Separate weighted linear model for each movie j:

$$\hat{y}_{ij} = m_j + e_i * \sum_{j_2 \in J_i} w_{j_2}$$

SVD-based methods with fewer parameters

Two models with $O(MK)$ parameters (M movies, K features):

$$\hat{y}_{ij} = c_i + d_j + e_i \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} w_{j_2 k}$$

$$\hat{y}_{ij} = c_i + d_j + \sum_{k=1}^{K} v_{jk} \sum_{j_2 \in J_i} v_{j_2 k}$$

Predictor	Test RMSE with basic predictors	Test RMSE with basic p. and with SVD with biases	Cumulative test RMSE
six basic predictors	.9826	.9039	.9826
regularized SVD	.9094	.9018	.9094
reg. SVD with biases	.9039	.9039	.9018
K-means	.9410	.9029	.9010
postproc. SVD with 1-NN	.9525	.9013	.8988
postproc. SVD with KRR	**.9006**	**.8959**	.8933
weighted linear models	.9506	.8995	.8902
small SVD-based 1	.9312	.8986	.8887
small SVD-based 2	.9590	.9032	.8879

Table: Linear regression results - RMSE on the test set

Experimental results

- the method with the best results: postprocessing SVD with kernel ridge regression
- combining all methods from the table plus 2 two-way interactions gives 6.34% improvement on the validation set (qualifying.txt)
- the 7.04% solution is a result of combining 56 predictors and 63 two-way interactions
- running times varied from 45min to 20h on a PC with 2GHz processor and 1.2GB RAM

Conclusions

- combining many methods with linear regression on the test set is a very effective approach
- good results of postprocessing SVD with kernel ridge regression suggest possibility of developing better methods than SVD for collaborative filtering